REMARKS

The present Amendment is responsive to the Official Action of September 26, 2007. The application contains five independent claims, Claims 1, 8, 18, 22, and 29. All of the independent claims (as well as the dependent claims) were rejected in the Official Action. All of the rejections are herein respectfully traversed. Reconsideration of all of the claims of the present application in light of the following remarks is respectfully requested.

I. Independent Claim 1

Independent Claim 1 reads as follows:

> 1. A method for protecting an operating system, comprising:
> determining integrity data for an operating system binary, wherein the integrity data enables detection of a modification to the operating system binary; and
> modifying a kernel with the integrity data, wherein the kernel is operable to employ the integrity data to detect the modification to the operating system binary.

The Official Action rejects this under 35 U.S.C. § 103(a) as being obvious over U.S. Patent Number 6,412,069 to Kavsan ("*Kavsan*") in view of U.S. Published Patent Application Number 2003/0120935 to Teal *et al.* ("*Teal*").

Kavsan discloses cryptographic service software that performs cryptographic services in the kernel space of an operating system. Such cryptographic services are generally useful, *Kavsan* explains, for encrypting data at the driver level, rather than at the application level as taught by the prior art. *Teal* is directed to a system and method for network security using a kernel based network security infrastructure. The system and method involve the installation of a computer code set into the operating system kernel of each computer on a network and use of the computer code set to detect and stop unwanted or malicious intrusions into the kernel.

The Official Action indicates that *Kavsan* "teaches a method for protecting an operating system, comprising: determining integrity data for an operating system binary, wherein the integrity data enables detection of a modification to the operating system binary (Column 2: lines 10-24; Column 2, lines 61-67; Column 3: lines 5-15, 20-27) . . ."

The cited portions of *Kavsan* read as follows:

> In accordance with one form of the present invention, cryptographic service software is embodied in at least one of a hard disc, a floppy disc or a read-only memory (ROM). The cryptographic service software electronically communicates and is compatible with a standard operating system of a computer, such as MicroSoft Windows (™). The operating system includes an application space and a kernel space. The cryptographic service software performs cryptographic services at the kernel space of the operating system. The cryptographic service software includes a generic layer having a kernel space level program interface, and a cryptographic service module having a library of encryption algorithms. This module may be replaced with a different module having updated or at least different encryption algorithms.
>
> . . .
>
> The cryptographic service software is compatible and communicates with a standard operating system of a computer, such as the Windows (™) operating system. Unlike the CryptoAPI (™) software, the cryptographic service software of the present invention is situated in the kernel space of the operating system, at the driver level of the computer. The cryptographic service software performs cryptographic services using encryption algorithms and the like at the kernel space of the operating system.
>
> The cryptographic service software is structured similarly to that of the CryptoAPI (™) software. It includes a generic layer having a kernel space level program interface 8, which functions and operates in a manner similar to the application program interface of the CryptoAPI (™) software. It further includes a cryptographic service module 10 which may be embodied in a similar manner to that of the CryptoAPI (™) software. The cryptographic service module 10 preferably includes a library of encryption algorithms. The module electronically communicates and cooperates with the kernel application programming interface 8. This module may be replaced with a different module having new or different encryption algorithms.
>
> The cryptographic service software allows one to write code at the driver level of the computer in a manner similar to the way the CryptoAPI (™) software does at the higher, application level. Now, encryption algorithms may be used to encrypt signals at the driver level, such as at the Ethernet port or at the modem port, video card or disk drive, etc., that is, at a level where the conventional

> CryptoAPT (™) software cannot reach. The cryptographic service software at the driver level is still accessible by application software 12 through secured drivers (engines) 14 situated at the driver level. Also, advantageously, during software development, the cryptographic software code at the kernel level may be debugged at the application level.

Nowhere in the cited passages does *Kavsan* disclose "determining integrity data for an operating system binary, wherein the integrity data enables detection of a modification to the operating system binary" as recited in Claim 1. The cited passages do indicate that the cryptographic service software is "situated in the kernel space of the operating system," "electronically communicates and is compatible with a standard operating system," and "performs cryptographic services . . . at the kernel space of the operating system." However, none of these is equivalent to or indicative of the determination of integrity data for the OS (*i.e.*, based on the OS binary system itself). Further, the cited passages indicate that "encryption algorithms may be used to encrypt signals at the driver level, such as at the Ethernet port or at the modem port, video card or disk drive, etc." As such, *Kavsan* is directed to the encryption, at the kernel level, of signals and not the OS binary.

This understanding of *Kavsan* is consistent with the Background of *Kavsan*, where it states (*See* Col. 1, ll. 54-61):

> Kernel space routines cannot cross the line into application space very efficiently and use the services of CryptoAPI (™) software in the application space. Therefore, if one wants to encrypt data or instructions coming in or out of the hard drive, the CryptoAPI (™) software would not be usable, as it resides in the application space and not in the kernel space. Similarly, the IP packets would also not be able to be encrypted using the CryptoAPI (™) software, as the IP packets are processed in the kernel space.

This passage makes clear that *Kavsan* is intended to address the need for encryption of data being communicated at the kernel space level and not the issue of determining the integrity of the data comprising the OS itself.

It is also noted that the preceding Official Action (dated January 16, 2007) admitted that *Kavsan* did not disclose "determining integrity data and detection of a modification to the operating system binary." As such, it is unclear why the Examiner has altered this position in the present Official Action.

The Official Action goes on to state (*see* p. 3) that *Kavsan* "does not explicitly disclose modifying a kernel with . . . integrity data. *Teal* in analogous art, however, disclose modifying a kernel with the integrity data (0035; 0044; 0067; 0091; 0095; 0097)."

Reviewing the cited passages of *Teal*, these passages, and indeed all of *Teal*, are directed to methods for preventing the insertion of malicious programming code into an operating system. This objective is achieved in part through the modification of the operating system achieved by loading a computer code set into the kernel space that enables identification and prevention of attempts to insert unwanted computer instructions into the kernel space and/or through the kernel space into one or more user spaces. However, *Teal* nowhere discusses systems or methods for detecting modifications to the OS, and does not disclose modifying a kernel with integrity data, which integrity data enables detection of a modification to the operating system binary.

For at least any of the above reasons, Claim 1, and the claims depending therefrom, are patentable over *Kavsan* and *Teal*, taken either alone or in combination.

II. Independent Claims 8, 18, 22, and 29

Independent Claims 8, 18, 22, and 29 were rejected under 35 U.S.C. § 103(a) as being obvious over International Published Patent Application Number WO 01/80482 to Eun *et al.* ("*Eun*") in view of *Teal*. Each of the claims at issue includes the limitations (albeit in slightly varied forms): "retrieving the first integrity data for an operating system binary" and "performing a tamper detection action if the first integrity data indicates tampering of the operating system binary."

Teal is discussed above. *Eun* is directed to a method and apparatus for protecting a file system. In general, *Eun* discloses the use of a digital signature-based authentication process in which a user seeking access to a computer file system is identified through a digital signature to determine whether the user is authorized for the sought access.

The Official Action states (*see* p. 6) that *Eun* discloses "a method for protecting an operating system, comprising . . . retrieving the first integrity data associated with the operating system binary (Figure 3: 312, 314 318) . . . and performing a tamper detection action if the first integrity data indicates tampering of the operating system binary (Figure 3: 310 308 306)." However, as pointed out in Applicants' prior response of July 16, 2007, the digital signature-based authentication of *Eun* is unrelated to the determining of OS tampering, and does not involve "integrity data" (defined in the present application as data that enables detection of a modification of the OS binary). (Applicants note that, due to the fact that the Examiner utilized a new reference (*Teal*) in the present Official Action in rejecting Claims 8, 18, 22, and 29, this prior argument was deemed "moot" and was not addressed in the present Official Action.) This deficiency in *Eun* is not cured by *Teal*, which, as discussed above, does not disclose the detection of OS modifications or the use of OS-related "integrity data."

For at least these reasons, independent Claims 8, 18, 22, and 29, and the claims depending therefrom, are patentable over *Eun* and *Teal*, taken either alone or in combination.

CONCLUSION

In view of the amended claims and the foregoing remarks, it is respectfully submitted that all of the claims of the present application are in condition for immediate allowance. It is therefore respectfully requested that a Notice of Allowance be issued. The Examiner is encouraged to contact Applicant's undersigned attorney to resolve any remaining issues in order to expedite examination of the present application.

It is not believed that extensions of time or fees for net addition of claims are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 CFR § 1.136(a), and any fee required therefore (including fees for net addition of claims) is hereby authorized to be charged to Deposit Account No. 16-0605.

Respectfully submitted,

/Richard D. Emery/

Richard D. Emery
Registration No. 58,894

Customer No. 00826
ALSTON & BIRD LLP
Bank of America Plaza
101 South Tryon Street, Suite 4000
Charlotte, NC 28280-4000
Tel Charlotte Office (704) 444-1000
Fax Charlotte Office (704) 444-1111
LEGAL02/30632599v1

ELECTRONICALLY FILED USING THE EFS-WEB ELECTRONIC FILING SYSTEM OF THE UNITED STATES PATENT & TRADEMARK OFFICE ON DECEMBER 18, 2007.